



LIBRARY
OF THE
UNIVERSITY
OF ILLINOIS

510.84

Iℓ6r

no.349-354

cop.2



Digitized by the Internet Archive
in 2013

<http://archive.org/details/erroranalysisofs351lerm>

510.84
IL62
no 351
cop. 2

math

Report No. 351

AN ERROR ANALYSIS OF SOLUTIONS TO SPARSE
LINEAR PROGRAMMING PROBLEMS

by

Raymond J. Lermitt

September 17, 1969

OCT 21 1969



DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS

University of Illinois
OCT 20 1969
Libran

Report No. 351

AN ERROR ANALYSIS OF SOLUTIONS TO SPARSE
LINEAR PROGRAMMING PROBLEMS*

By

Raymond J. Lermitt

September 17, 1969

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

*This work was supported in part by the Advanced Research Projects Agency as administered by the Rome Air Development Center under Contract No. US AF 30(602)4144 and submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science, September, 1969.

ABSTRACT

This thesis discusses the round-off errors incurred in solving Linear Programming problems by the Revised Simplex Method, using the Product Form of the Inverse - the most usual method of solution. The matrices involved are assumed to be sparse as is usually the case.

Also discussed are the reinversion of the basis matrix and iterative refinement of the solution obtained.

ACKNOWLEDGEMENT

The author wishes to express his sincere gratitude to his adviser and friend, Professor Robert S. Northcote, for his help in preparing this thesis. He would also like to thank his co-workers Serena Yardley and Teruji Yamamoto.

A special word of thanks to Mrs. Kay Flessner for her patient work in typing a very difficult manuscript.

TABLE OF CONTENTS

	Page
1. INTRODUCTION	1
2. NOTATION AND PRELIMINARY RESULTS	3
3. THE SIMPLEX METHOD	10
4. ERROR ANALYSIS OF THE PRODUCT FORM OF THE INVERSE METHOD	15
5. REINVERSION OF THE BASIS MATRIX	30
6. ERROR ANALYSIS OF THE REINVERSION ROUTINE	38
7. ITERATIVE REFINEMENT	49
8. CONCLUDING REMARKS	55
LIST OF REFERENCES	57
APPENDIX	58

1. INTRODUCTION

The history of linear programming goes back to the discovery of the simplex algorithm in 1947 by George L. Dantzig, and its implementation on digital computers in the mid 1950's. Little, however, has been written about the round-off errors produced during the computation, the most notable exception being a thesis by Bartels [1].

A linear programming problem may be defined as:

minimize a linear objective function

$$z = c^T x$$

subject to the constraints

$$Ax = b, x \geq 0,$$

where A is an $m \times n$ real matrix and x , b and c are real vectors of length n , m and n , respectively.

Two approaches to error analysis are possible. The forward error analysis method assumes that a solution $x + \delta x$ has been obtained and the size of δx is then estimated. Alternatively, the backward error analysis approach assumes that the solution obtained is the exact solution of the slightly different problem:

$$\text{minimize: } z + \delta z = (c + \delta c)^T x$$

$$\text{subject to: } (A + \delta A) x = b + \delta b, x \geq 0$$

and estimates of the size of δA , δb and δc are made.

Even the most cursory glance at input data for a linear programming problem will convince the reader that the information is far from exact.

For this reason backward error analysis is a much more realistic approach and is the one which will be pursued in this paper. If a bound can be given on the sizes of δA , δb and δc , and these values are within tolerances determined by inaccuracies in the data, then the problem may be considered solved. This places on the user the responsibility of specifying the accuracy of his data.

A feature of nearly all linear programming problems is that the matrix A is sparse, with very few, often less than one percent, of its elements differing from zero. This fact has been used almost from the outset to reduce the amount of computation and storage required simply by not storing the zeros and omitting calculations involving them. It is also possible to exploit this sparseness in the error analysis.

Control of errors in linear programming has traditionally been done by "tolerances". Quantities that are "very small" are set to zero and numbers greater than some small negative constant are deemed to be positive. Such tolerances are usually set by "the system" but may be changed by the user at his peril.

Efficient solution of problems seems to require these tolerances, and techniques for their automatic control have been developed (e.g., Clasen [2]). However, the final solution should be free of any such artificialities and this is easily achieved by iteratively refining an approximate solution to purify it of its tainted past.

Chapters 3 and 4 cover the simplex method and its backward error analysis respectively. Chapters 5 and 6 cover a reinversion routine that concentrates on retaining sparseness and providing error control. If the solution is not good enough after reinversion the refinement techniques of Chapter 7 may be used. Finally, Chapter 8 shows how slight imperfections in the solution may be removed by perturbing the original problem.

2. NOTATION AND PRELIMINARY RESULTS

It is assumed that all operations are performed in normalized floating-point arithmetic. Define

$$fl(E)$$

to be the result of evaluating the expression E . Following Wilkinson [9], the following laws in floating point arithmetic are assumed to hold on a normal machine, providing overflow and underflow do not occur:

$$fl(a + b) = a(1 + \delta_1) + b(1 + \delta_2)$$

$$fl(a - b) = a(1 + \delta_3) - b(1 + \delta_4)$$

$$fl(a \times b) = (a \times b) (1 + \delta_5)$$

$$fl(a / b) = (a / b) (1 + \delta_6)$$

for any two machine numbers a and b , the quantities $|\delta_i|$, which are relative errors, being bounded by some small constant.

Two degrees of precision are utilized:

normal precision, in which

$$|\delta_i| \leq \epsilon_1$$

and extended precision where

$$|\delta_i| \leq \epsilon_2$$

No assumptions are made about the relative sizes of ϵ_1 and ϵ_2 . Normal precision is used throughout, except for iterative refinement. Note, however, that many routines use "double precision" for all calculations.

It is assumed that operations involving a zero operand are exact; i.e.,

$$\text{fl}(0 + a) = \text{fl}(a - 0) = a$$

$$\text{fl}(0 - a) = -a$$

$$\text{fl}(0 \times a) = \text{fl}(a \times 0) = 0$$

$$\text{fl}(0 / a) = 0 \quad (a \neq 0)$$

In practice, however, careful programming should be used to avoid doing these operations at all, as they are redundant.

Upper case letters are used for matrices, lower case letters are used for vectors and for the elements of both matrices and vectors. Where a sequence of matrices or vectors is used, superscripts are employed to distinguish them and their elements are similarly labeled. Matrices and vectors of small terms introduced to account for rounding errors are prefixed by δ i.e., δB is a perturbation applied to the matrix B . All vectors are column vectors unless the context implies otherwise.

Throughout the analysis, the ℓ_∞ vector norm

$$\|v\| = \max_i |v_i|$$

is used and

$$\begin{aligned} \|A\| &= \max_{\|x\|=1} \|Ax\| \\ &= \max_i \sum_{j=1}^n |a_{ij}| \end{aligned}$$

for any $n \times n$ matrix A . Since sparse matrices are being used, it is convenient to define the function ∇ by

$$\nabla(t) = \begin{cases} 1 & \text{if } t \neq 0 \\ 0 & \text{if } t = 0 \end{cases}$$

for any scalar t . Thus the number of nonzero elements in a vector

$v = (v_1, v_2, \dots, v_n)$ is

$$\sum_{i=1}^n \nabla(v_i)$$

Lemma

If $|\theta_k| \leq \epsilon \quad k = 1, 2, \dots, i$

then

$$\left| \prod_{k=1}^i (1 + \theta_k) - 1 \right| \leq i \epsilon'$$

where

$$\epsilon' = \frac{1}{n} \left[\exp \left(\frac{n \epsilon}{1 - \epsilon} \right) - 1 \right] \quad \text{and} \quad i \leq n \quad (1)$$

Proof:

$$\begin{aligned} \text{Let } p &= \prod_{k=1}^i (1 + \theta_k) \\ &\geq \prod_{k=1}^i (1 - \epsilon) = (1 - \epsilon)^i \end{aligned} \quad (2)$$

$$\begin{aligned} p &\leq (1 + \epsilon)^i \\ &\leq \left(\frac{1}{1 - \epsilon} \right)^i = (1 - \epsilon)^{-i} \end{aligned}$$

$$p - 1 \leq (1 - \epsilon)^{-i} - 1 \quad (3)$$

Since

$$- \left[(1 - \epsilon)^{i/2} - (1 - \epsilon)^{-i/2} \right]^2 \leq 0$$

$$1 - (1 - \epsilon)^i \leq (1 - \epsilon)^{-i} - 1$$

Hence, from (1),

$$1 - p \leq (1 - \epsilon)^i - 1 \quad (4)$$

Now (2) and (3) together yield

$$\begin{aligned} |p - 1| &\leq (1 - \epsilon)^{-i} - 1 \\ &= \left(1 + \frac{\epsilon}{1 - \epsilon} \right)^i - 1 \\ &\leq \frac{i}{n} \left[\exp \left(\frac{n \epsilon}{1 - \epsilon} \right) - 1 \right] \end{aligned}$$

For small ϵ , ϵ' is only slightly larger than ϵ . It is often convenient to use ϵ' instead of ϵ in error bounds.

In the evaluation of the inner product of two dense vectors, it is necessary to estimate the error in calculating

$$p = \text{fl} \left(\sum_{i=1}^n x_i y_i \right)$$

using floating-point arithmetic. Let

$$s_1 = 0$$

$$s_{i+1} = \text{fl} (s_i + x_i y_i) \quad i = 1, 2, \dots, n$$

$$= s_i (1 + \theta_i) + x_i y_i (1 + \Delta_i)^2$$

$$\text{where } |\theta_i| \leq \epsilon, \quad |\Delta_i| \leq \epsilon$$

$$\text{and } \epsilon = \text{either } \epsilon_1 \text{ or } \epsilon_2$$

Then

$$p = s_{n+1}$$

$$= \sum_{i=1}^n x_i y_i (1 + \Delta_i)^2 \prod_{k=i+1}^n (1 + \theta_k)$$

Using the lemma,

$$p = \sum_{i=1}^n x_i y_i [1 + (n - i + 2) \gamma_i]$$

$$\text{where } |\gamma_i| \leq \epsilon'$$

$$= \sum_{i=1}^n x_i y_i (1 + \delta_i) \quad , \quad |\delta_i| \leq (n + 1) \epsilon'$$

Now consider the case where x and y are sparse vectors. Let

$$q_x = \sum_{i=1}^n \nabla (x_i)$$

$$q_y = \sum_{i=1}^n \nabla (y_i)$$

$$q = \min (q_x, q_y)$$

and

$$q' = \sum_{i=1}^n \nabla (x_i y_i)$$

Then

$$q' \leq q$$

If X_j, Y_j are x_i, y_i , respectively, where $x_i y_i$ is the j -th nonzero term in the sum, then

$$\begin{aligned}
fl \left(\sum_{i=1}^n x_i y_i \right) &= fl \left(\sum_{j=1}^{q'} X_j Y_j \right) \\
&= \sum_{j=1}^{q'} X_j Y_j (1 + \Delta_j) \\
&\quad \text{where } |\Delta_j| \leq (q' + 1)\epsilon' \\
&= \sum_{i=1}^n x_i y_i (1 + \delta_i) \\
&\quad \text{where } |\delta_i| \leq (q+1)\epsilon'
\end{aligned}$$

In fact, $\delta_i = \Delta_j$; however, use of the less stringent bound is justified, since q' is hard to estimate but q is easily obtained.

It is sometimes convenient to consider one of the vectors (say y) to be dense, in this case:

$$q_y = n$$

$$q = q_x$$

so

$$fl \left(\sum_{i=1}^n x_i y_i \right) = \sum_{i=1}^n x_i y_i (1 + \delta_i)$$

where

$$|\delta_i| \leq (q_x + 1) \epsilon'.$$

3. THE SIMPLEX METHOD

A brief outline of a version of the Revised Simplex algorithm is given; a much more detailed analysis may be found in Dantzig [3] or Hadley [6].

The problem to be solved may be put into the form:

minimize

$$z = c^T x$$

subject to

$$Ax = b$$

$$x \geq 0.$$

Let B be the basis matrix $[P_{v_1}, \dots, P_{v_m}]$ where P_{v_1}, \dots, P_{v_m} is a set of m linearly independent columns of A . Then a basic solution is given by:

$$\hat{x} = B^{-1} b,$$

where
$$\hat{x} = [x_{v_1}, \dots, x_{v_m}]^T,$$

since the inverse exists; this solution is also feasible if $x \geq 0$. The success of the Simplex method depends on the fact that, if an optimal solution exists, there is a basic feasible solution which yields the optimum value of the objective function. It is therefore necessary to consider only the basic feasible solutions in the search for optimality.

An algorithm for solving the problem by the Revised Simplex Method is as follows:

- (a) Find an initial basic feasible solution, or indicate that none exists.

- (b) Change the basis until optimality is reached, which is achieved by a series of simplex iterations. The steps required for each iteration, starting with some basis $B = [P_{v_1}, P_{v_2}, \dots, P_{v_m}]$ are:

Step 1: Set $\gamma^T = [c_{v_1}, c_{v_2}, \dots, c_{v_m}]$.

Step 2: Compute $\pi^T = \gamma^T B^{-1}$.

Step 3: Compute $d_j = c_j - \pi^T P_j$

for $j = 1 (1) n, j \notin \{v_1, v_2, \dots, v_m\}$.

Choose an index s such that

$$d_s = \min \{d_j : d_j < 0\}$$

If no $d_j < 0$, the solution is optimal.

Step 4: Compute $y = B^{-1} P_s$

Step 5: Find an index r such that

$$\frac{x_r}{y_r} = \min_{y_i > 0} \left\{ \frac{x_i}{y_i} \right\}$$

If no y_i is > 0 , then the solution is unbounded

Choosing the minimum of x_i/y_i assures that the new basic solution x' will be feasible.

Step 6: Change the basis to

$$B' = [P_{v_1}, \dots, P_{v_{r-1}}, P_s, P_{v_{r+1}}, \dots, P_{v_m}].$$

Step 7: Compute the new solution, x' , using

$$x' = (B')^{-1} b.$$

Since three sets of equations are solved using the matrix B , it is advantageous to have B^{-1} available. If B^{-1} is known, then the inverse of the new basis $(B')^{-1}$ is easily obtained, since

$$\begin{aligned} (B')^{-1} &= (B')^{-1} B B^{-1} \\ &= (B^{-1} B')^{-1} B^{-1}. \end{aligned}$$

B' is identical to B except that the r -th column, P_{v_r} , is replaced by P_s , so $B^{-1} B'$ takes on the simple form:

$$B^{-1} B' = \begin{bmatrix} 1 & & & & y_1 & & & & \\ & 1 & & & y_2 & & & & \\ & & \ddots & & \vdots & & & & \\ & & & \bigcirc & \vdots & & & & \\ & & & & \vdots & & & \bigcirc & \\ & & & & \vdots & & & & \\ & & & & \vdots & & & & \\ & & & & 1 & & & & \\ & & \bigcirc & & y_r & & & & \\ & & & & \vdots & 1 & & & \\ & & & & \vdots & & \ddots & & \\ & & & & \vdots & & & \bigcirc & \\ & & & & y_m & & & & 1 \end{bmatrix}$$

$$\text{where } y = [y_1, y_2, \dots, y_m]^T = B^{-1} P_s.$$

Such a matrix is called an elementary column matrix, or simply an elementary matrix. It is easily inverted to give:

$$(B^{-1}B')^{-1} = \begin{bmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & \ddots & & & & \\ & & & 1 & & & \\ & & & & 1/y_r & & \\ & & & & \vdots & & \\ & & & & & 1 & \\ & & & & & & \ddots & \\ & & & & & & & 1/y_r & \\ & & & & & & & & \ddots & \\ & & & & & & & & & 1 & \\ & & & & & & & & & & \ddots & \\ & & & & & & & & & & & 1 \end{bmatrix} = T$$

say, and

$$(B')^{-1} = T B^{-1}$$

Clearly, if the starting basis is B_0 , and T^1, T^2, \dots, T^p are the elementary matrices produced during p iterations, then the inverse of the current basis B_p , say, is given by:

$$B_p^{-1} = T^p T^{p-1} \dots T^2 T^1 B_0^{-1}$$

It is usual to take $B_0 = B_0^{-1} = I$, so that

$$B_p^{-1} = T^p T^{p-1} \dots T^2 T^1.$$

If a unit matrix cannot be found from the columns of A , then one must be manufactured using "artificial" columns. When the corresponding artificial variables are all removed (or reduced to zero) by subsequent iterations, a basic feasible solution has been found.

When the matrix B_p^{-1} is not calculated explicitly, but only T^p , T^{p-1} , ..., T^2 , T^1 are stored, the method is called the product form of the inverse. We will restrict our analysis to this method, since it is used more extensively than any other method in large scale LP algorithms.

In order to reduce round off errors, storage requirements and running time, it is desirable from time to time to start afresh and recalculate the inverse of the current basis. This technique, called reinversion, is discussed in Chapter 5.

4. ERROR ANALYSIS OF THE PRODUCT FORM OF THE INVERSE METHOD

In this chapter the errors introduced during one iteration of the Simplex method are considered.

Suppose that, after p iterations, the approximate inverse of the basis B is given in product form by:

$$T^p T^{p-1} \dots T^2 T^1$$

where each T^j is an elementary matrix. Define an error matrix E by:

$$T^p T^{p-1} \dots T^2 T^1 (B + E) = I. \quad (1)$$

Suppose also that, in performing one iteration, the r -th column of B is replaced by a non-basis column of A , $A_s = w$ say, giving a new basis matrix B' . A new elementary matrix T^{p+1} is formed which satisfies:

$$T^{p+1} T^p T^{p-1} \dots T^2 T^1 (B' + E') = I \quad (2)$$

The round off errors in (1) and (2) are measured by $||E||$ and $||E'||$, respectively. The problem is to find a bound for $||E'||$ in terms of $||E||$.

To simplify the notation assume, without loss of generality, that the j -th elementary matrix, T^j , is a unit matrix except for its j -th column and that in the j -th iteration, it is the j -th column of B that is replaced. Thus, pivoting occurs down the diagonal. This can be achieved by suitable interchange of rows and columns. In practice the actual pivots may be recorded by saving the pivot column number corresponding to each component of the basic solution. Thus set the j -th elementary matrix to a unit matrix except for the j -th column which is

$$[\eta_1^j, \eta_2^j, \dots, \eta_m^j]^T$$

Since T^j is obtained by inverting an elementary matrix whose j -th column is y^j say, define the vector η^j to be given exactly by

$$\eta_i^j = \begin{cases} -y_i^j / y_j^j & i = 1(1)m, i \neq j \\ 1 / y_j^j & i = j. \end{cases}$$

Whenever the T^j is used in calculation, the error introduced by the divisions will be included in the error analysis.

Either of two different methods may be used in calculating η^j , in each of which $t = -y_j^j$ and $y_j^j = -1$ are set first. One method is then to divide each element of y^j by t . The other is to multiply each element by t^{-1} . The latter method will be faster if division takes longer than multiplication; it does, however, produce two round off errors for each element. It is assumed that the latter method is used.

One step in the Simplex algorithm is to compute

$$y = B^{-1} w$$

for which the approximate solution is given by

$$y = f1 (T^p T^{p-1} \dots T^2 T^1 w) \quad (3)$$

Setting

$$\alpha^1 = w$$

and calculating successively

$$\alpha^{k+1} = \text{fl} (T^k \alpha^k) \quad (k = 1, 2, \dots, p) \quad (4)$$

yields

$$y = \alpha^{p+1}.$$

Equation (4) is equivalent to

$$\alpha_i^{k+1} = \begin{cases} \text{fl} (\alpha_i^k + \eta_i^k \alpha_k^k) & i=1, 2, \dots, m \\ & i \neq k \\ \text{fl} (\eta_k^k \alpha_k^k) & i = k \end{cases} \quad (5)$$

Rearrangement of (5) to include the effect of computations on previous α^k gives:

$$\alpha_i^\ell = \begin{cases} \text{fl} (\sum_{k=i}^{\ell-1} \alpha_k^k \eta_i^k) & , \quad i < \ell \\ \text{fl} (w_i + \sum_{k=1}^{\ell-1} \alpha_k^k \eta_i^k) & , \quad i \geq \ell \end{cases} \quad (6)$$

In particular, setting $i = \ell$, yields

$$\alpha_\ell^\ell = \text{fl} (w_\ell + \sum_{k=1}^{\ell-1} \alpha_k^k \eta_\ell^k). \quad (7)$$

For convenience in notation, let $\beta_k = \alpha_k^k$. Then

$$\beta_\ell = \text{fl} (w_\ell + \sum_{k=1}^{\ell-1} \beta_k \eta_\ell^k) \quad (8)$$

and

$$y_i = \alpha_i^{p+1} = \begin{cases} f1 \left(\sum_{k=i}^p \beta_k \eta_i^k \right) & i \leq p \\ f1 \left(w_i + \sum_{k=1}^p \beta_k \eta_i^k \right) & i > p \end{cases} \quad (9)$$

Now let L be the matrix

$$L = \begin{bmatrix} 1 & & & & \\ -\eta_2^1 & 1 & & & \\ -\eta_3^1 & -\eta_3^2 & 1 & & \\ \vdots & \vdots & & \ddots & \\ -\eta_m^1 & -\eta_m^2 & \dots & \dots & 1 \end{bmatrix}$$

Then (8) may be written in the form:

$$L \beta = w$$

from which β is uniquely given by

$$\beta = L^{-1} w.$$

Using the relations given in Lemma 1 of the Appendix, it can be seen that β exactly satisfies

$$(L + \delta L) \beta = w + \delta w ,$$

where

$$(\delta L)_{ij} = \begin{cases} \eta_i^j \delta_{ij} & i > j \\ 0 & i \leq j \end{cases},$$

$$|\delta_{ij}| \leq (q_i + 3) \epsilon_1',$$

$$q_i = \sum_{k=1}^{i-1} \nabla (\eta_i^k),$$

$$|\delta w_i| \leq q_i |w_i| \epsilon_1'.$$

$$\text{If } \Delta w = L \beta - w$$

$$\text{then } \Delta w = \delta w - \delta L \cdot \beta$$

$$\begin{aligned} \therefore ||\Delta w|| &\leq ||\delta w|| + ||\delta L|| ||\beta|| \\ &\leq q_L ||w|| \epsilon_1' + ||\delta L|| ||\beta|| \end{aligned} \quad (10)$$

$$\text{where } q_L = \max_i q_i.$$

$$\text{If } M_L = \max_{1 \leq j \leq i \leq m} |\eta_i^j| \quad (11)$$

$$\text{then } ||\delta L|| \leq \epsilon_1' q_L (q_L + 3) M_L \quad (12)$$

and hence

$$||\Delta w|| \leq \epsilon_1' (q_L ||w|| + q_L (q_L + 3) M_L ||\beta||) \quad (13)$$

Now estimate the error involved in calculating y using (9).

Let U be the matrix

$$U = \begin{bmatrix} \eta_1^1 & \eta_1^2 & \dots & \eta_1^p & & \\ & \eta_2^2 & \dots & \eta_2^p & & \\ & & \ddots & \vdots & & \\ & & & \eta_{p-1}^{p-1} & \eta_{p-1}^p & \\ & & & & \eta_p^p & \\ \eta_{p+1}^1 & \eta_{p+1}^2 & \dots & \eta_{p+1}^p & 1 & \\ \vdots & & & \vdots & & \ddots \\ \eta_m^1 & \eta_m^2 & \dots & \eta_m^p & & & 1 \end{bmatrix}$$

provided that $p \leq m$. Then

$$y = f1 (U \gamma)$$

$$\text{where } \gamma = [\beta_1, \beta_2, \dots, \beta_p, w_{p+1}, \dots, w_m]^T \quad (14)$$

By using the results of Lemma 2 in the Appendix, it can be shown that y satisfies

$$y = U (\gamma + \delta \gamma)$$

where

$$\delta \gamma = [\delta \beta_1, \delta \beta_2, \dots, \delta \beta_p, \Delta w_{p+1}, \dots, \Delta w_m]^T \text{ say,}$$

$$\begin{aligned}
 ||\delta \gamma|| &\leq \frac{||U^{-1}||}{1 - 2 \epsilon'_1 q_U(q_U+2) M_U ||U^{-1}||} \\
 &\times \left[\epsilon'_1 q_U(q_U+2) M_U ||y|| + \epsilon'_1 ||\gamma|| \right] \quad (15)
 \end{aligned}$$

$$q_U = \max_i \begin{cases} \sum_{k=i}^p \nabla (\eta_i^k) & , i \geq p \\ \sum_{k=1}^p \nabla (\eta_i^k) + 1 & , i < p \end{cases}$$

= the maximum number of nonzero elements in any row of U.

and

$$M_U = \max_{i,j} (|U_{ij}|).$$

$$\text{If } \delta \beta = (\delta \beta_1, \delta \beta_2, \dots, \delta \beta_p)$$

$$\text{and } \Delta w' = (\Delta w_{p+1}, \dots, \Delta w_n)$$

$$\text{then } ||\delta \beta|| \leq ||\delta \gamma|| \text{ and } ||\Delta w'|| \leq ||\delta \gamma||$$

The perturbation $\delta \beta$ of β gives rise to a perturbation $L \delta \beta$ in w_1, \dots, w_p

Therefore, combining (13) and (15)

$$\begin{aligned}
 y &= f1 (T^p T^{p-1} \dots T^2 T^1 w) \\
 &= T^p T^{p-1} \dots T^2 T^1 (w + v)
 \end{aligned}$$

where

$$||v|| \leq \max \{ \epsilon'_1 (q_L ||w|| + q_L (q_L + 3) M_L ||\beta||) + ||L|| ||\delta\beta||, ||\delta\gamma|| \} \quad (16)$$

and

$$||L|| \leq q_L M_L$$

To simplify (16), let

$$M = \max_{i,k} (|\eta_i^k|, 1) \quad (17)$$

Then

$$M_L \leq M, M_U \leq M. \quad (18)$$

Let

$$q = \max_i \sum_{k=1}^p \nabla (\eta_i^k)$$

Then

$$||v|| \leq \epsilon'_1 (q ||w|| + q (q+3) M ||\beta||) + \frac{\epsilon'_1 q M ||U^{-1}||}{1 - 2 \epsilon'_1 q (q+3) M ||U^{-1}||} [q (q+3) M ||y|| + ||\beta|| + q ||w||] \quad (19)$$

The only quantity in the above inequality which is unknown is $||U^{-1}||$; fortunately approximations to the elements of U^{-1} are found during the calculation of the η vectors.

Consider now the calculation of the η vectors forming the approximate inverse of β , and consider β to be a unit matrix except for the first p columns. If

$$B = \begin{bmatrix} \alpha_{11}^1 & \alpha_{12}^1 & \cdot & \cdot & \cdot & \cdot & \alpha_{1p}^1 \\ \alpha_{21}^1 & \alpha_{22}^1 & \cdot & \cdot & \cdot & \cdot & \alpha_{2p}^1 \\ \vdots & & & & & & \vdots \\ \alpha_{p1}^1 & \alpha_{p2}^1 & \cdot & \cdot & \cdot & & \alpha_{pp}^1 \\ \vdots & & & & & & \vdots \\ \alpha_{m1}^1 & \alpha_{m2}^1 & \cdot & \cdot & \cdot & & \alpha_{mp}^1 \end{bmatrix} \begin{matrix} \\ \\ \\ \\ 1 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{matrix}$$

and

$$\alpha_{ij}^{k+1} = \begin{cases} f1 (\alpha_{ij}^k + \alpha_{kj}^k \eta_i^k) & i \neq k \\ f1 (\alpha_{kj}^k \eta_k^k) & i = k \end{cases} \quad (20)$$

for $i = 1(1)m$, $j = 1(1)p$, $k = 1(1)j-1$

then

$$\alpha_{ij}^j = \begin{cases} f1 (\sum_{k=i}^{j-1} \alpha_{kj}^k \eta_i^k) & i < j \\ f1 (\alpha_{ij}^1 + \sum_{k=1}^{j-1} \alpha_{kj}^k \eta_i^k) & i \geq j \end{cases} \quad (21)$$

and the η vectors are given by

$$\eta_i^j = \begin{cases} f1 (-\alpha_{ij}^j / \alpha_{jj}^j) & i \neq j \\ f1 (1 / \alpha_{jj}^j) & i = j \end{cases} \quad (22)$$

Let Γ be the matrix

$$\begin{bmatrix} \alpha_{11}^1 & \alpha_{12}^1 & \alpha_{13}^1 & \dots & \alpha_{1p}^1 \\ & \alpha_{22}^2 & \alpha_{23}^2 & \dots & \alpha_{2p}^2 \\ & & \alpha_{33}^3 & & \vdots \\ & & & \ddots & \vdots \\ & & & & \alpha_{pp}^p \\ \alpha_{p+1,1}^1 & \alpha_{p+1,2}^1 & \dots & \alpha_{p+1,p}^1 & 1 \\ \vdots & \vdots & & \vdots & \vdots \\ \alpha_{m1}^1 & \alpha_{m2}^1 & \dots & \alpha_{mp}^1 & 1 \end{bmatrix}$$

Then

$$(U\Gamma)_{ij} = \begin{cases} \sum_{k=i}^j \alpha_{kj}^k \eta_i^k & i \leq p, j \leq p \\ \sum_{k=1}^i \alpha_{kj}^k \eta_i^k + \alpha_{ij}^1 & i > p, j \leq p \\ 1 & i = j > p \\ 0 & \text{otherwise} \end{cases} \quad (23)$$

But, for $i < j \leq p$,

$$\sum_{k=i}^j \alpha_{kj}^k \eta_i^k = \alpha_{jj}^j \eta_i^j + \sum_{k=i}^{j-1} \alpha_{kj}^k \eta_i^k$$

and, from (21),

$$\alpha_{ij}^j = \sum_{k=i}^{j-1} \alpha_{kj}^k \eta_i^k + \sum_{k=i}^{j-1} \delta_{ij}^k \alpha_{kj}^k \eta_i^k$$

where

$$| \delta_{ij}^k | \leq (q_i + 2) \epsilon'_1$$

$$q_i = \sum_{k=1}^p \nabla (\eta_i^k)$$

$$\therefore \sum_{k=i}^j \alpha_{kj}^k \eta_i^k = \alpha_{ij}^j + \alpha_{jj}^j \eta_i^j + \sum_{k=i}^{j-1} \delta_{ij}^k \alpha_{kj}^k \eta_i^k. \quad (24)$$

Similarly, for $i > p$,

$$\begin{aligned} \sum_{k=1}^i \alpha_{kj}^k \eta_i^k + \alpha_{ij}^1 &= \alpha_{ij}^j + \alpha_{jj}^j \eta_i^j + \sum_{k=1}^{j-1} \delta_{ij}^k \alpha_{kj}^k \eta_i^k \\ &\quad + \Delta_{ij} \alpha_{ij}^1 \end{aligned} \quad (25)$$

but, from (22),

$$\alpha_{ij}^j + \alpha_{jj}^j \eta_i^j = 2 \delta'_{ij} \alpha_{ij}^j \quad i < j \quad (26)$$

$$\text{and} \quad \alpha_{jj}^j \eta_j^j = 1 + \alpha_{jj}^j \eta_j^j \delta_{jj}^j$$

where

$$| \delta'_{ij} | \leq \epsilon'_1$$

Hence

$$UT = I + F$$

where F is a matrix of small terms caused by round off errors.

$$\text{If } q = \max_i \sum_{k=1}^p \nabla (\eta_i^k)$$

$$M = \max_i (|\eta_i^k|, 1),$$

as before, G is the matrix

$$G = \begin{bmatrix} \alpha_{11}^1 & \alpha_{12}^1 & \alpha_{13}^1 & \dots & \alpha_{1p}^1 \\ & \alpha_{22}^2 & \alpha_{23}^2 & \dots & \alpha_{2p}^2 \\ & & \ddots & & \vdots \\ & & & \alpha_{p-1, p-1}^{p-1} & \alpha_{p-1, p}^{p-1} \\ & & & & \alpha_{pp}^p \end{bmatrix}$$

and H is a $p \times p$ matrix given by

$$H_{ij} = \alpha_{ij}^j$$

Then

$$F_{ij} = \begin{cases} \alpha_{jj}^j \eta_j^j \delta_{jj}^j & i = j \leq p \\ \sum_{k=i}^{j-1} \alpha_{kj}^k \eta_i^k \delta_{ij}^k + 2 \alpha_{ij}^j \delta_{ij}' & i < j \leq p \\ \sum_{k=1}^{j-1} \alpha_{kj}^k \eta_i^k \delta_{ij}^k + 2 \alpha_{ij}^j \delta_{ij}' + \alpha_{ij}^1 \Delta_{ij} & i > p, j \leq p \\ 0 & \text{otherwise} \end{cases}$$

Hence

$$\begin{aligned}
 \sum_{j=1}^m |F_{ij}| &\leq \sum_{j=1}^p \left(\left| \sum_{k=1}^j \delta_{ij}^k \alpha_{kj}^k \right| + |\eta_i^k| \right) \\
 &\quad + 2\epsilon'_1 \sum_{j=1}^p |\alpha_{ij}^j| + \epsilon'_1 \sum_{j=1}^p |\alpha_{ij}^1| \\
 &\leq \epsilon'_1 [q(q+2)M ||G|| + 2 ||H|| + ||B||]
 \end{aligned}$$

Thus

$$||F|| \leq \epsilon'_1 [q(q+2)M ||G|| + 2 ||H|| + ||B||] \quad (27)$$

and

$$||R|| \leq \max \{ ||G||, ||B|| + 1 \}$$

Now

$$U^{-1} = R(I + F)^{-1}$$

$$\therefore ||U^{-1}|| \leq ||R|| ||(I + F)^{-1}||$$

$$\leq \frac{||R||}{1 - ||F||} \quad \text{provided } ||F|| < 1 \quad (28)$$

Substituting the bound for $||U^{-1}||$ into equation (19) gives a bound for $||v||$.

It has been shown that

$$\begin{aligned} y &= fl (T^p T^{p-1} \dots T^2 T^1 w) \\ &= T^p T^{p-1} \dots T^2 T^1 (w + v). \end{aligned}$$

It is now possible to estimate $||E'||$, where E' satisfies:

$$T^{p+1} T^p \dots T^2 T^1 (B' + E') = I.$$

The new elementary matrix T^{p+1} has as its $p + 1$ -st column the vector η^{p+1} where:

$$\eta_i^{p+1} = \begin{cases} -y_i / y_{p+1} & i \neq p+1 \\ 1 / y_{p+1} & i = p+1 \end{cases}$$

Thus

$$\begin{aligned} T^{p+1} y &= [\overbrace{0, \dots, 0}^{p \text{ zeros}}, 1, 0, \dots, 0]^T \\ &= T^{p+1} T^p \dots T^2 T^1 (w + v) \end{aligned}$$

since w is the $p + 1$ -st column of B' , v is the $p+1$ -st column of E' .

For all other columns B' and B are identical, and, since

$$\begin{aligned} T^{p+1} \dots T^2 T^1 (B + E) &= T^{p+1} I \\ &= T^{p+1}, \end{aligned}$$

$(B' + E')$ and $(B + E)$ are the same except for column $p+1$; i.e., E' is E with the $p+1$ -st column replaced by v . Hence

$$||E'|| \leq ||E|| + ||v||$$

Finding a bound for $||v||$ is not as difficult as equations (19) and (27) would seem to indicate since the only quantities required that are not recorded are the elements of G . (β is just the last column of G). These values are obtained during the calculation of y and may be used to update $||G||$ from iteration to iteration.

This bound depends heavily on:

1. The sparseness of the η vectors;
2. The size of their elements.

It is, therefore, very important to reduce the magnitude of the η elements by controlling the size of 'pivots'. Consideration also should be given to maintaining sparseness.

5. REINVERSION OF THE BASIS MATRIX

During the execution of the simplex algorithm, the choice of pivot is restricted at each iteration by the need to obtain a significant improvement in the value of the objective function and to maintain feasibility. Little can be done to control round-off errors and the sparseness of the inverse. In recalculating the inverse a flexible pivot selection agenda can be adopted, and the errors and density be thereby greatly reduced.

Before discussing the general reinversion algorithm, the decomposition of a sparse matrix A by two different methods will be considered. In both cases pivoting occurs down the diagonal.

(1) Product form decomposition.

$$A = E^1 E^2 \dots E^{n-1} E^n$$

where

$$E^j = \begin{bmatrix} 1 & & & & & & & & \\ & 1 & & & e_{1j} & & & & \\ & & \ddots & & e_{2j} & & & & \\ & & & \ddots & \vdots & & & & \\ & & & & 1 & & & & \\ & & & & & e_{j-1,j} & & & \\ & & & & & e_{jj} & & & \\ & & & & & e_{j+1,j} & 1 & & \\ & & & & & \vdots & & \ddots & \\ & & & & & e_{nj} & & & 1 \end{bmatrix}$$

(2) Decomposing A into a product of lower and upper triangular matrices

$$L = \begin{bmatrix} l_{11} & & & & \\ l_{21} & l_{22} & & & \\ l_{31} & l_{32} & l_{33} & & \\ \vdots & \vdots & \vdots & \ddots & \\ \vdots & \vdots & \vdots & & \ddots \\ l_{n1} & l_{n2} & l_{n3} & \dots & l_{nn} \end{bmatrix}$$

and

$$U = \begin{bmatrix} 1 & u_{12} & u_{13} & \dots & u_{1n} \\ & 1 & u_{23} & \dots & u_{2n} \\ & & \ddots & & \vdots \\ & & & \ddots & \vdots \\ & & & & 1 & u_{n-1,n} \\ & & & & & 1 \end{bmatrix}$$

so that $A = LU$.

Having decomposed the matrix, inversion is easy, since

$$(E^j)^{-1} = \begin{bmatrix} 1 & & & -e_{1j}/e_{jj} & & \\ & 1 & & \vdots & & \\ & & \ddots & \vdots & & \\ & & & 1/e_{jj} & & \\ & & & \vdots & 1 & \\ & & & & & \ddots \\ & & & -e_{nj}/e_{jj} & & & 1 \end{bmatrix}$$

$$L^{-1} = L^n L^{n-1} \dots L^2 L^1$$

where

$$L^j = \begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ & & \ddots & & & \\ & & & 1 & & \\ & & & & 1/\ell_{jj} & \\ & & & & & 1 \dots 1 \\ & & & -\ell_{j+1,j}/\ell_{jj} & & \\ & & & \vdots & & \\ & & & -\ell_{nj}/\ell_{jj} & & \end{bmatrix}$$

and similarly for U^{-1} . Note that the pattern of zeros and nonzeros in E^j , L^j , U^j correspond exactly to that of the matrices from which they were derived.

The density of the product form decomposition is at least as great as in the LU form, as can be shown thus:

$$E^j = \begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ & & \ddots & & & \\ & & & 1 & & \\ & & & & e_{jj} & \\ & & & & e_{j+1,j} & 1 \\ & & & & \vdots & \ddots \\ & & & & e_{nj} & \ddots & 1 \end{bmatrix} \begin{bmatrix} 1 & & & & e_{1j} & \\ & 1 & & & \vdots & \\ & & \ddots & & e_{j-1,j} & \\ & & & 1 & 1 & \\ & & & & 1 & 1 \dots 1 \end{bmatrix}$$

$$= S^j T^j \text{ (say).}$$

$$\text{Then } A = S^1 T^1 S^2 T^2 \dots S^{n-1} T^{n-1} S^n T^n.$$

$$\text{For } i \leq j, \quad T^i S^j = S^j T^i$$

So the product of the S's and T's may be rearranged to give:

$$A = S^1 S^2 \dots S^{n-1} S^n T^1 T^2 \dots T^{n-1} T^n.$$

S^1, S^2, \dots, S^n are all lower triangular matrices, so $L = S^1 S^2 \dots S^n$ is lower triangular.

Similarly $U = T^1 T^2 \dots T^n$ is upper triangular, and LU is the (unique) decomposition into lower and upper triangular parts.

$$L = S^1 S^2 \dots S^n$$

$$= \begin{bmatrix} e_{11} & & & \\ e_{21} & 1 & & \\ \vdots & & \ddots & \\ e_{n1} & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & e_{22} & & \\ & e_{23} & 1 & \\ & \vdots & & \ddots \\ & e_{n2} & & & 1 \end{bmatrix} \dots \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & e_{n-1,n-1} \\ & & & e_{nn-1,n-1} & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 & \\ & & & & e_{nn} \end{bmatrix}$$

$$= \begin{bmatrix} e_{11} & & & \\ e_{21} & e_{22} & & \\ e_{31} & e_{32} & e_{33} & \\ \vdots & \vdots & \vdots & \ddots \\ e_{n1} & e_{n2} & e_{n3} & \dots & e_{nn} \end{bmatrix}$$

Hence L contains the same number of nonzero elements as in S^1, S^2, \dots, S^n (excluding unit diagonals which need not be stored).

$$U = T^1 T^2 \dots T^n$$

$$\therefore U^{-1} = (T^n)^{-1} (T^{n-1})^{-1} \dots (T^2)^{-1} (T^1)^{-1}$$

$$\begin{aligned}
&= \begin{bmatrix} 1 & & & & -e_{1n} \\ & 1 & & & -e_{2n} \\ & & \ddots & & \vdots \\ & & & \ddots & -e_{n-1n} \\ & & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & & -e_{1n-1} \\ & 1 & & & -e_{2n-1} \\ & & \ddots & & \vdots \\ & & & \ddots & 1 \\ & & & & 1 \end{bmatrix} \cdots \begin{bmatrix} 1 & -e_{12} \\ & 1 \\ & & \ddots \\ & & & 1 \end{bmatrix} \\
&= \begin{bmatrix} 1 & -e_{12} & -e_{13} & \cdots & -e_{1n} \\ & 1 & -e_{23} & \cdots & -e_{2n} \\ & & 1 & \cdots & -e_{3n} \\ & & & \ddots & \vdots \\ & & & & 1 & -e_{n-1,n} \\ & & & & & 1 \end{bmatrix}
\end{aligned}$$

Therefore the number of nonzero elements in T^1, T^2, \dots, T^n is equal to that in U^{-1} (again excluding the unit diagonals).

Since the density of U^{-1} will be greater than that of U , due to "fill in" during the inversion procedure, it may be concluded that the decomposition into elementary matrices will, in general, give rise to more nonzero elements than will LU decomposition. (This ignores the possibility of cancellation, which might produce zero elements from sums of nonzero ones; however, this should not often happen in practice).

It is therefore desirable to use an LU decomposition method for reinversion, not only to reduce the rounding errors, but also to reduce the number of η elements representing the inverse.

It has been shown that a triangular matrix decomposes readily into elementary matrices. It is therefore desirable to sort the rows and columns of the basis to form a matrix which is as close to a triangular matrix as possible. The sorted basis is then partitioned as follows:

	R	M	C	N
R				
M				
C				
N				

In practice the matrix need not be physically sorted, all that is needed are the indices of the pivot positions.

The NN block contains logical vectors in the basis; these will produce no η records. The RR and CC blocks are triangular. The MM block contains rows and columns which cannot be triangularized by sorting. LU decomposition is applied on this block.

Rewrite the matrix, in partitioned form, as:

R	O		
	M	O	O
S		C	
	N	D	I

where O and I are zero and identity matrices of appropriate sizes. This decomposes to:

$$\begin{array}{|c|c|c|c|} \hline R & O & & \\ \hline & I & O & O \\ \hline S & O & I & \\ \hline & & O & I \\ \hline \end{array} \times \begin{array}{|c|c|c|c|} \hline I & & O & \\ \hline & M & & O \\ \hline O & N & I & \\ \hline & & O & I \\ \hline \end{array} \times \begin{array}{|c|c|c|c|} \hline I & O & & \\ \hline & I & O & \\ \hline O & O & C & \\ \hline & & D & I \\ \hline \end{array}$$

The first and last matrices are lower triangular, pivoting down the diagonal gives η vectors with no increase in the number of nonzero elements.

If $M = LU$, L and U being lower and upper triangular respectively, then

$$\begin{array}{|c|c|c|c|} \hline I & O & & \\ \hline & M & O & O \\ \hline O & N & I & \\ \hline & & O & I \\ \hline \end{array} = \begin{array}{|c|c|c|c|} \hline I & O & & \\ \hline & L & O & O \\ \hline O & O & I & \\ \hline & & O & I \\ \hline \end{array} \times \begin{array}{|c|c|c|c|} \hline I & O & & \\ \hline & U & & O \\ \hline O & N & I & \\ \hline & & O & I \\ \hline \end{array}$$

The first matrix is lower triangular and the second can be transformed by row and column interchanges into:

$$\begin{array}{|c|c|c|c|} \hline I & O & & O \\ \hline & I & O & \\ \hline O & & I & N \\ \hline & O & O & U \\ \hline \end{array}$$

which is upper triangular. After inversion, the interchanges can be reversed. The only place where elements are added to the inverse is in the formation of L and U. Since the pivoting strategy within the M block may be employed, this may be used to minimize the number of nonzeros created.

The exact form of the pivoting strategy depends on such factors as the number of columns of the matrix that can be stored in memory at one time. An important point is to choose the pivots dynamically, at each step choosing that pivot which minimizes the number of nonzeros created.

It may be necessary to change the pivot selection method where choosing a particular pivot would result in very large elements in the representation of the inverse. Discussion of this point is deferred until the next chapter.

6. ERROR ANALYSIS OF THE REINVERSION ROUTINE

In this chapter the computation of the inverse of the matrix

$$B = \begin{array}{|c|c|c|c|} \hline R & O & & \\ \hline & M & O & \\ \hline S & & C & \\ \hline & N & & \\ \hline & & D & I \\ \hline \end{array}$$

$\underbrace{\hspace{1.5cm}}_r$
 $\underbrace{\hspace{1.5cm}}_s$
 $\underbrace{\hspace{1.5cm}}_t$
 columns columns columns

using the product form

$$T^p T^{p-1} \dots T^2 T^1, \quad p = r + s + t,$$

is considered. It is shown that

$$T^p T^{p-1} \dots T^2 T^1 (B + \delta B) = I$$

i.e., $T^p \dots T^1$ is the exact inverse of some perturbed matrix $B + \delta B$, and a bound for $||\delta B||$ is established. As before B is split into the form

where $|\delta_{ij}| \leq \epsilon_1$,

The terms $b_{ij} (1 + \delta_{ij})$ may be regarded as the elements of a matrix; hence $T^r \dots T^1$ is the exact inverse of the perturbed matrix $(B_R + \delta B_R)$ where

$$(\delta B_R)_{ij} = \begin{cases} \delta_{ij} (B_R)_{ij} & j = i(1) r \\ 0 & \text{otherwise} \end{cases}$$

Similarly, the product of the elementary matrices produced for the inversion of B_C is the exact inverse of $(B_C + \delta B_C)$ where

$$(\delta B_C)_{ij} = \begin{cases} \delta_{ij} (B_C)_{ij} & j = r + s + 1(1)p \\ 0 & \text{otherwise} \end{cases}$$

and $|\delta_{ij}| \leq \epsilon_1$

This leaves only the matrix

$$B_M = \begin{array}{|c|c|c|c|} \hline I & O & & \\ \hline & M & O & O \\ \hline O & & I & \\ \hline & N & O & I \\ \hline \end{array}$$

to be inverted. Since the elements of N are simply copied into the η vectors with a change in sign, the only error is that in inverting M . Therefore set,

$$M=M^1 = \begin{bmatrix} m_{11}^1 & m_{12}^1 & \cdot & \cdot & \cdot & m_{1s}^1 \\ m_{21}^1 & m_{22}^1 & \cdot & \cdot & \cdot & m_{2s}^1 \\ \cdot & \cdot & & & & \cdot \\ \cdot & \cdot & & & & \cdot \\ \cdot & \cdot & & & & \cdot \\ m_{s2}^1 & m_{s2}^1 & \cdot & \cdot & \cdot & m_{ss}^1 \end{bmatrix}$$

and compute a sequence of matrices $M^1, M^2, \dots, M^s, M^{s+1}$, by forming a new η vector from the k -th column of M^k using only elements on or below the diagonal.

Thus

$$\eta_k^k = \text{fl} (1/m_{kk}^k) = \frac{1}{m_{kk}^k (1+\delta_{kk}^k)}$$

$$\eta_i^k = \text{fl} (-m_{ik}^k \eta_k^k) = \frac{-m_{ik}^k (1+\delta_{ik}^k)}{m_{kk}^k (1+\delta_{kk}^k)}$$

$$\text{where } |\delta_{ik}^k| \leq \epsilon_1'. \quad (1)$$

The matrix M^k is then updated by this new elementary matrix giving M^{k+1} . If there were no rounding errors, the elements of M^{k+1} would be zero in the k -th column below the diagonal and m_{kk}^{k+1} would be unity. If we set these to zero and one respectively then:

$$m_{ij}^{k+1} = \begin{cases} 1 & i = j = k \\ 0 & i > k, j = k \\ f_1(\eta_k^k m_{kj}^k) & i = k, j > k \\ f_1(m_{ij}^k + \eta_i^k m_{kj}^k) & i > k, j > k \\ m_{ij}^k & \text{otherwise} \end{cases} \quad (2)$$

i.e.,

$$m_{ij}^{k+1} = \begin{cases} 1 & i = j = k \\ 0 & i > k, j = k \\ \eta_k^k m_{kj}^k + \eta_k^k m_{kj}^k \delta_{kj}'^k & i = k, j > k \\ m_{ij}^k + \eta_i^k m_{kj}^k + m_{ij}^k \delta_{ij}^k + 2 \eta_i^k m_{kj}^k \delta_{ij}'^k & i > k, j > k \\ m_{ij}^k & \text{otherwise} \end{cases} \quad (3)$$

Thus if

$$L^k = \begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ & & \ddots & & & \\ & & & 1 & & \\ & & & \eta_k^k & & \\ & & & \eta_{k+1}^k & 1 & \\ & & & \vdots & & \ddots \\ & & & \eta_p^k & & & 1 \end{bmatrix}$$

and E^k is the matrix with elements e_{ij}^k ,

then

$$M^{k+1} = L^k M^k + E^k \quad (4)$$

where

$$\epsilon_{ij}^k = \begin{cases} m_{kk}^k \eta_k^k \delta_{kk}^k & i = j = k \\ m_{ik}^k \delta_{ik}^k + m_{kk}^k \eta_i^k \delta_{kk}^k & i > k, j = k \\ m_{kj}^k \eta_k^k \delta_{kj}^k & i = k, j > k \\ m_{ij}^k \delta_{ij}^k + 2 m_{kj}^k \eta_i^k \delta_{ij}^k & i > k, j > k \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

and $|\delta_{ij}^k| \leq \epsilon_1^k$, $|\delta_{ij}^k| \leq \epsilon_1^k$, for all i, j, k .

After s iterations $M^{s+1} = U$, (say), an upper triangular matrix which has ones down the diagonal. Let

$$\begin{aligned} F^k &= (L^k)^{-1} E^k \\ \therefore M^{k+1} &= L^k M^k + L^k F^k \\ &= L^k (M^k + F^k) \end{aligned}$$

But the elements of F^k are zero except for $i \geq k$ and $j \geq k$ (because those of E^k are).

$$\begin{aligned} \therefore L^i F^k &= F^k \quad \text{for } i < k \\ \therefore L^{k-1} L^{k-2} \dots L^2 L^1 F^k &= F^k \end{aligned}$$

$$\begin{aligned}
\therefore U &= M^{s+1} \\
&= L^s L^{s-1} \dots L^2 L^1 M^1 + \sum_{k=1}^s (L^s L^{s-1} \dots L^k F^k) \\
&= L^s L^{s-1} \dots L^2 L^1 M^1 + \sum_{k=1}^s (L^s L^{s-1} \dots L^2 L^1 F^k) \\
&= L^s L^{s-1} \dots L^2 L^1 (M + F) \dots \dots \dots (6)
\end{aligned}$$

where

$$F = \sum_{k=1}^s F^k.$$

If

$$U = \begin{bmatrix} 1 & u_{12} & u_{13} & \dots & u_{1s} \\ & 1 & u_{23} & & u_{2s} \\ & & 1 & \dots & u_{3s} \\ & & & \ddots & \vdots \\ & & & & 1 & u_{s-1s} \\ & & & & & 1 \end{bmatrix}$$

then

$$U^{-1} = U^2 U^3 \dots U^{s-1} U^s$$

where

$$U^j = \begin{bmatrix} 1 & & & & -u_{1j} \\ & \cdot & & & -u_{2j} \\ & & \cdot & & \vdots \\ & & & \cdot & \vdots \\ & & & & -u_{j-1j} \\ & & & & 1 \\ & & & & \cdot \\ & & & & \cdot \\ & & & & \cdot \\ & & & & 1 \end{bmatrix}$$

from which

$$U^2 U^3 \dots U^{s-1} U^s L^s L^{s-1} \dots L^2 L^1 (M+F) = I.$$

A bound on $\|F\|$ is now required. Since

$$F^k = (L^k)^{-1} E^k,$$

$$F_{ij}^k = \begin{cases} m_{ik}^k \delta_{ik}^k & i \geq k, j = k \\ m_{ij}^k \delta_{kj}^k & i = k, j > k \\ m_{ij}^k \delta_{ij}^k + 2 \eta_i^k m_{kj}^k (\delta_{ij}^{'k} - \delta_{kj}^k) & i > k, j > k \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

An important point is that these error terms do not involve the 'pivot' element η_k^k . Summing over k gives

$$F_{ij} = \begin{cases} m_{ij}^j \delta_{ij}^j + \sum_{k=1}^{j-1} (m_{ij}^k \delta_{ij}^k + 2 \eta_i^k m_{kj}^k (\delta_{ij}^{',k} - \delta_{kj}^k)) & i \geq j \\ m_{ij}^i \delta_{ij}^i + \sum_{k=1}^{i-1} (m_{ij}^k \delta_{ij}^k + 2 \eta_i^k m_{kj}^k (\delta_{ij}^{',k} - \delta_{kj}^k)) & i < j \end{cases} \quad (8)$$

$$\text{Let } \rho = \max_{i,j,k} \{ |m_{ij}^k| : i \geq k \}$$

$$M = \max_{i,k} \{ |\eta_i^k| \}$$

$$q_i = \sum_{k=1}^s \nabla (\eta_i^k)$$

$$\sigma_{ij} = \sum_{k=1}^s \nabla (m_{ij}^k)$$

$$\text{Then } |F_{ij}| \leq (\sigma_{ij} + 4q_i) \rho M \in_1'$$

$$\therefore ||F|| \leq s (\sigma + 4q) \rho M \in_1'$$

$$\text{where } q = \max_i (q_i)$$

$$\sigma = \max_{i,j} (\sigma_{ij}).$$

This gives an a posteriori bound for $||F||$ in terms of the sparseness and the size of the elements of the inverse and the intermediate matrices M^1, M^2, \dots, M^p . As might be expected, sparse matrices give much smaller errors.

Combining these results indicates that the computed inverse is the exact inverse, in product form, of the perturbed matrix

$$B + \delta B = \begin{array}{|c|c|c|c|} \hline R+\delta R & O & & \\ \hline & M+F & O & \\ \hline S+\delta S & N & C+\delta C & \\ \hline & & D+\delta D & I \\ \hline \end{array}$$

It is clear from equation (9) that both sparseness and size of the elements are crucial factors in bounding the error. The previous chapter dealt with preserving sparseness. Consideration now must be given to bounding the size of the elements in the η vectors and the intermediate matrices that are formed. In forming an η vector, start with some column vector $v = [v_1, v_2, \dots, v_p]^T$ (say) and 'pivot' on some element v_r , giving

$$\eta = \text{fl} [-v_1/v_r, \dots, -v_{r-1}/v_r, 1/v_r, -v_{r+1}/v_r, \dots, -v_p/v_r]^T$$

where the value of r may be arbitrarily chosen. The usual method is to select the pivot element so that $|v_r|$ is greater than some 'tolerance' value so that the elements of η will not be 'too large'. Since, as equation (7) shows, the element $1/v_r$ does not appear in any error term, the remaining terms may

be bounded by c (say) by insisting that r be chosen such that:

$$|v_r| \geq \frac{1}{c} |v_i| \quad \text{for } i = 1, \dots, r-1, \\ r+1, \dots, p$$

It may be noted that for dense systems, one normally selects the largest element of v for this pivot, which is equivalent to taking $c = 1$. Thus the above scheme is a compromise between minimizing the number of nonzero elements and reducing their size.

7. ITERATIVE REFINEMENT

Suppose that an approximate inverse to the basis B and an approximate solution vector x have been calculated. It is desired to calculate a more accurate solution using iterative refinement. A detailed analysis of iterative refinement using floating point arithmetic is given by Moler [7]. A simple adaption of that work is developed here to cover the sparse matrix case.

Each iteration of the refinement requires the following three steps:

- 1) Compute the residuals: $r^m = b - Bx^m$ with $x^1 = x$.
- 2) Compute the correction: $d^m = B^{-1}r^m$.
- 3) Add the correction to form: $x^{m+1} = x^m + d^m$.

The step where most accuracy is required is in calculating the residuals.

It is assumed that extended precision is used in this step. Computing $d^m = B^{-1}r^m$ is done using the approximate inverse to B .

Let

$$r^m = b - Bx^m = c^m \quad (1)$$

$$d^m = (B + E^m)^{-1} r^m \quad (2)$$

$$x^{m+1} = x^m + d^m + g^m \quad (3)$$

where c^m , E^m and g^m are correction terms due to round-off error and are introduced to make the equations hold exactly. A bound for $||E^m||$ is known, using the results of either Chapter 4 or Chapter 6. It is required that the inverse be reasonably good so that

$$||E^m|| < (||B^{-1}||)^{-1}$$

Let

$$F^m = B^{-1} E^m.$$

Then

$$||F^m|| = ||B^{-1} E^m|| \leq ||B^{-1}|| ||E^m|| < 1.$$

and hence $I + F^m$ is nonsingular.

Equation (2) becomes

$$d^m = (I + F^m)^{-1} B^{-1} r^m. \quad (4)$$

Suppose x is the exact solution $x = B^{-1} b$.

Then

$$\begin{aligned} x^{m+1} - x &= x^m - x + d^m + g^m \\ &= (I + F^m)^{-1} [(I + F^m)(x^m - x) + B^{-1}(b - Bx^m + c^m)] + g^m \\ &= (I + F^m)^{-1} [F^m(x^m - x) + B^{-1} c^m] + g^m \end{aligned}$$

Hence

$$||x^{m+1} - x|| \leq \frac{||F^m||}{1 - ||F^m||} ||x^m - x|| + \frac{||B^{-1}||}{1 - ||F^m||} (||c^m|| + ||g^m||)$$

Suppose an upper bound for $||E^m||$ is given by $||\delta B||$. Then

$$||x^{m+1} - x|| \leq \frac{||B^{-1}||}{1 - ||\delta B|| ||B^{-1}||} (||\delta B|| ||x^m - x|| + ||c^m||) + ||g^m||$$

Estimates of $||c^m||$ and $||g^m||$ are now required.

$$g^m = fl(x^m + d^m) - (x^m + d^m)$$

$$\therefore g_i^m = \theta_i x_i^m + \theta'_i d_i^m, \text{ for } |\theta_i| \leq \epsilon_1, |\theta'_i| \leq \epsilon_1$$

$$\therefore ||g^m|| \leq \epsilon_1 (||x^m|| + ||d^m||)$$

When the process converges, $||d^m|| \leq ||x^m||$

$$\therefore ||g^m|| \leq 2 \epsilon_1 ||x^m||$$

$$\leq 2 \epsilon_1 (||x|| + ||x^m - x||).$$

In calculating r^m , it is assumed that each component $r_i^m = fl_2(b_i - \sum B_{ij} x_j^m)$ is calculated using extended precision and the result then is rounded to normal precision. This is denoted by $fl_{2,1}(b_i - \sum B_{ij} x_j^m)$.

$$c^m = r^m - b + Bx^m$$

and

$$\begin{aligned}
c_i^m &= f_{1,2,1} (b_i - \sum B_{ij} x_j^m) - (b_i - \sum B_{ij} x_j^m) \\
&= (f_{1,2} (b_i - \sum B_{ij} x_j^m)) (1 + \theta_i) - (b_i - \sum B_{ij} x_j^m) \\
&= \{b_i(1+\xi_i) - \sum B_{ij} x_j^m (1 + \gamma_{ij})\} (1 + \theta_i) - (b_i - \sum B_{ij} x_j^m) \\
&= (b_i - \sum B_{ij} x_j^m) \theta_i \\
&\quad + \{b_i \xi_i - \sum B_{ij} x_j^m \gamma_{ij}\} (1 + \theta_i)
\end{aligned}$$

where $|\theta_i| \leq \epsilon_1, |\xi_i| \leq \epsilon_2$

$$|\gamma_{ij}| \leq (q + 2) \epsilon'_2$$

$$q = \max_i \sum_{j=1}^m \nabla (B_{ij})$$

$$\begin{aligned}
\therefore ||c^m|| &\leq \epsilon'_1 ||b - Bx^m|| + (1 + \epsilon_1) \epsilon'_2 ||b|| \\
&\quad + (q + 2) (1 + \epsilon_1) \epsilon'_2 ||B|| ||x^m|| \\
&\leq \epsilon'_1 ||B|| ||x^m - x|| + (1 + \epsilon_1) \epsilon'_2 ||B|| ||x|| \\
&\quad + (q + 2) (1 + \epsilon_1) \epsilon'_2 ||B|| (||x|| + ||x^m - x||)
\end{aligned}$$

$$\begin{aligned}
\text{since } ||x^m|| &= ||x + (x^m - x)|| \\
&\leq ||x|| + ||x^m - x||.
\end{aligned}$$

$$\begin{aligned}
\therefore ||c^m|| &\leq [\epsilon'_1 + (q + 2) (1 + \epsilon_1) \epsilon'_2] ||B|| ||x^m - x|| \\
&\quad + (q + 3) (1 + \epsilon_1) \epsilon'_2 ||B|| ||x||.
\end{aligned}$$

$$\begin{aligned}
\therefore ||x^{m+1} - x|| &\leq \frac{||\delta B|| ||B^{-1}||}{1 - ||\delta B|| ||B^{-1}||} ||x^m - x|| \\
&\quad + \frac{||\delta B|| ||B|| ||B^{-1}||}{1 - ||\delta B|| ||B^{-1}||} \left[\begin{aligned} &\{\epsilon'_1 + (q + 2) (1 + \epsilon_1) \epsilon'_2\} ||x^m - x|| \\ &+ (q + 3) (1 + \epsilon_1) \epsilon'_2 ||x|| \end{aligned} \right] \\
&\quad + 2 \epsilon_1 (||x|| + ||x^m - x||)
\end{aligned}$$

$$= \left[\frac{||\delta B|| ||B^{-1}|| + k(B) \{\epsilon'_1 + (q+2)(1+\epsilon_1)\epsilon'_2\} + 2\epsilon_1}{1 - ||\delta B|| ||B^{-1}||} ||x^m - x|| \right] \\ + \left[\frac{k(B)}{1 - ||\delta B|| ||B^{-1}||} (q+3)(1+\epsilon_1)\epsilon'_2 + 2\epsilon_1 \right] ||x||$$

where $k(B) = ||B|| ||B^{-1}||$, the condition number of B .

$$= \sigma_1 ||x^m - x|| + \sigma_2 ||x|| \quad (\text{say}).$$

$$\therefore \frac{||x^{m+1} - x||}{||x||} \leq \sigma_1 \frac{||x^m - x||}{||x||} + \sigma_2$$

Since x^1 is calculated in the same way (using, in effect, $x^0 \equiv 0$)

$$||x^1 - x|| \leq \sigma_1 ||x||$$

(This may be shown by setting $\epsilon'_2 = 0$ for this step (no calculation of residuals is involved) and noting that the only other term in σ_2 , $2\epsilon_1$, came from $2\epsilon_1 ||x^m|| \leq 2\epsilon (||x|| + ||x^m - x||)$, but $x^0 \equiv 0$).

$$\therefore \frac{||x^1 - x||}{||x||} \leq \sigma_1$$

\therefore by induction,

$$\frac{||x^m - x||}{||x||} \leq \sigma_1^m + \frac{\sigma_2}{1 - \sigma_1}, \quad \sigma_1 \leq 1.$$

Therefore, provided $\sigma_1 < 1$, convergence of x^m to within the tolerance given by

$$\frac{\sigma_2}{1-\sigma_1} ||x||,$$

can be guaranteed.

σ_1 depends essentially on $||\delta B||$, $||B^{-1}||$ and the measure of sparseness, q . It is essential that B be well conditioned and that $||\delta B||$ be small, i e., an accurate inverse of B is known.

The effect of a small value of q is to reduce the need for using double precision in calculating the residuals, since it is always multiplied by ϵ'_2 .

8. CONCLUDING REMARKS

The question remains as to what constitutes an optimal feasible solution. Strictly speaking, it is one for which $x \geq 0$ and the reduced costs d_j are also ≥ 0 for $j = 1(1)n$. In practice, small negative tolerances, δ_f , δ_o (say) are chosen so that the solution is considered optimal and feasible provided

$$x_i \geq \delta_f \quad \text{for all } i$$

$$d_j \geq \delta_o \quad \text{for all columns } j \text{ not in the basis.}$$

An alternative approach is as follows: Determine some sufficiently refined solution x and a basis B , such that

$$Bx = b$$

may be considered exact. If some x_i are < 0 , define

$$\bar{x} = x + \delta x \geq 0$$

where

$$\delta x_i = \begin{cases} 0 & , \quad x_i \geq 0 \\ -x_i & , \quad x_i < 0 \end{cases}$$

This produces a change in b , δb (say), where

$$\delta b = B \delta x$$

The vector δb can either be calculated directly, or its norm may be estimated from

$$||\delta b|| \leq ||B|| ||\delta x||$$

If the elements of δb are sufficiently small that $b + \delta b$ is an acceptable right hand side, then \bar{x} may be considered to be the required solution to the problem.

Similarly, in calculating $d_j = c_j - \sum_{i=1}^m \pi_i a_{ij}$, π may be refined in the same way as x , so that the d_j are accurately known. If $d_j < 0$ for some j , a small change in the corresponding c_j will set them to zero. If this new cost vector is acceptable, then the problem is solved.

The techniques described in Chapters 3 through 6 and above enable the exact solution to the slightly perturbed problem:

minimize

$$z + \delta z = (c + \delta c)^T x$$

subject to

$$(A + \delta A) x = b + \delta b$$

$$x \geq 0.$$

to be obtained. If $||\delta A||$, $||\delta b||$ and $||\delta c||$ are within acceptable limits, then the problem may be considered solved. If more accuracy is required, iterative refinement as described in Chapter 7 may be used.

LIST OF REFERENCES

- [1] Bartels, Richard H., "A Numerical Investigation of the Simplex Method", Technical report no. CS 104, Stanford University, 1968.
- [2] Clasen, R. J., "Techniques for Automatic Tolerance Control in Linear Programming", Comm. ACM, 9 (November 1966), pp. 802-803.
- [3] Dantzig, George B., Linear Programming and Extensions, Princeton University Press, 1965.
- [4] Dantzig, George B., Harvey, Roy P., McKnight, Robert D., Smith, Stanley S., "Sparse Matrix Techniques in two Mathematical Programming Codes", Technical report no. 69-1, Stanford University, 1969.
- [5] Forsythe, George E. and Moler, Cleve B., Computer Solution of Linear Algebraic Systems, Prentice Hall, 1967.
- [6] Hadley, G., Linear Programming, Addison Wesley, 1962.
- [7] Moler, Cleve B., "Iterative Refinement in Floating Point", JACM, 14 (April 1967), pp. 316-321.
- [8] Orchard-Hays, William, Advanced Linear Programming Techniques, McGraw Hill, 1968.
- [9] Wilkinson, J. H., Rounding Errors in Algebraic Processes, Prentice Hall, 1963.

APPENDIX

Lemma 1: The solution of a triangular system of equations.

Suppose that the components of a vector x are computed in the order x_1, x_2, \dots, x_n by

$$\begin{aligned} x_1 &= b_1 \\ x_i &= f1 \left(b_i + \sum_{k=1}^{i-1} r_{ik} x_k \right) \quad i = 2, 3, \dots, n \end{aligned}$$

or, equivalently, by the set of equations

$$\begin{bmatrix} 1 & & & & \\ -r_{21} & 1 & & & \\ -r_{31} & -r_{32} & 1 & & \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ -r_{m1} & -r_{m2} & & -r_{mm-1} & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix},$$

i.e., $Rx = b$, is solved. Then x is the exact solution of the perturbed triangular system:

$$(R + \delta R) x = b + \delta b,$$

where,

$$\delta R = \begin{bmatrix} 0 & & & & \\ -r_{21} \delta_{21} & 0 & & & \\ -r_{31} \delta_{31} & -r_{32} \delta_{32} & 0 & & \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ -r_{m1} \delta_{m1} & & -r_{mm-1} \delta_{mm-1} & 0 \end{bmatrix},$$

and

$$\begin{aligned}\delta^b &= [b_1 \delta'_1, \dots, b_m \delta'_m]^T, \\ |\delta'_i| &\leq q_i \epsilon'_1, \\ |\delta_{ik}| &\leq (q_i + 1) \epsilon'_1\end{aligned}$$

with

$$q_i = \sum_{k=1}^{i-1} \nabla (r_{ik})$$

Proof

In Chapter 2, it was shown that

$$\text{fl} \left(\sum_{i=1}^n x_i y_i \right) = \sum_{i=1}^n x_i y_i (1 + \delta_i)$$

where

$$|\delta_i| \leq (q_x + 1) \epsilon'$$

and

$$q_x = \sum_{i=1}^n \nabla (x_i)$$

Similarly, if $p = \text{fl} \left(z + \sum_{i=1}^n x_i y_i \right)$ is calculated from:

$$s_1 = z$$

$$s_{i+1} = \text{fl} (s_i + x_i y_i) \quad i = 1, 2, \dots, n$$

$$p = s_{n+1}$$

then

$$\text{fl} \left(z + \sum_{i=1}^n x_i y_i \right) = z (1 + \Delta) + \sum_{i=1}^n x_i y_i (1 + \delta_i)$$

where

$$| \Delta | \leq q_x \epsilon' ,$$

$$| \delta_i | \leq (q_x + 1) \epsilon'$$

and

$$q_x = \sum_{i=1}^n \nabla (x_i) .$$

Using this result,

$$\begin{aligned} x_i &= \text{fl} \left(b_i + \sum_{k=1}^{i-1} r_{ik} x_k \right) \\ &= b_i (1 + \delta'_i) + \sum_{k=1}^{i-1} r_{ik} x_k (1 + \delta_{ik}) \end{aligned}$$

where

$$| \delta_{ik} | \leq (q_i + 1) \epsilon'_1$$

and

$$| \delta'_i | \leq q_i \epsilon'_1 .$$

Hence

$$x_i = b_i + b_i \delta'_i + \sum_{k=1}^{i-1} (r_{ik} + r_{ik} \delta_{ik}) x_k$$

which completes the proof.

Note also that

$$\begin{aligned} ||\delta R|| &= \max_i \sum_{j=1}^{i-1} |r_{ij} \delta_{ij}| \\ &\leq \max_i (q_i \max_{1 \leq j \leq i-1} |r_{ij} \delta_{ij}|) \\ &\leq q (q+1) \epsilon'_1 \max_{i,j} |r_{ij}| \end{aligned}$$

$$\text{where } q = \max_i q_i.$$

If r_{ik} results from a division, as described earlier, two extra errors are involved so that, in the lemma,

$$|\delta_{ik}| \leq (q_i + 3) \epsilon'_1$$

$$\text{and } ||\delta R|| \leq q (q + 3) \epsilon'_1 \max_{i,j} |r_{ij}|.$$

Lemma 2. A direct error bound on x .

Under the same hypotheses as in Lemma 1, if \bar{x} is the exact solution $R^{-1}b$, and $\delta x = x - \bar{x}$, then

$$\|\delta x\| \leq \frac{\|R^{-1}\|}{1 - 2m\|R^{-1}\|} [m\|x\| + \epsilon'_1 \|b\|]$$

where $m = q(q+2) \epsilon'_1 \max_{i,k} |r_{ik}|$,

provided that

$$m \|R^{-1}\| < \frac{1}{2}.$$

Proof

As before,

$$(R + \delta R) x = b + \delta b$$

also $R \bar{x} = b$

$$\therefore b + \delta b = (R + \delta R) (\bar{x} - \delta x)$$

$$= R\bar{x} + \delta R\bar{x} - (R + \delta R) \delta x$$

$$\therefore \delta b = \delta R \bar{x} - (R + \delta R) \delta x$$

$$\therefore \delta x = (R + \delta R)^{-1} [\delta R \bar{x} - \delta b]$$

$$\text{and } ||\delta x|| \leq ||(R + \delta R)^{-1}|| [||\delta R|| ||\bar{x}|| + ||\delta b||]$$

$$\text{but } ||(R + \delta R)^{-1}|| = ||[R(I + R^{-1}\delta R)]^{-1}||$$

$$\leq ||R^{-1}|| ||(I + R^{-1}\delta R)^{-1}||$$

$$\leq \frac{||R^{-1}||}{1 - ||R^{-1}\delta R||}$$

$$\leq \frac{||R^{-1}||}{1 - ||R^{-1}|| ||\delta R||}$$

$$\text{provided } ||R^{-1}|| ||\delta R|| < 1$$

$$(\text{and hence } ||R^{-1}\delta R|| < 1)$$

$$\therefore ||\delta x|| \leq \frac{||R^{-1}||}{1 - ||R^{-1}|| ||\delta R||} [||\delta R|| ||\bar{x}|| + \epsilon'_1 ||b||]$$

$$\text{But } \bar{x} = x - \delta x,$$

$$\therefore ||\bar{x}|| \leq ||x|| + ||\delta x||$$

$$\therefore ||\delta x|| \leq \frac{||R^{-1}||}{1 - ||R^{-1}|| ||\delta R||} [||\delta R|| (||x|| + ||\delta x||) + \epsilon'_1 ||b||]$$

which reduces to

$$||\delta x|| \leq \frac{||R^{-1}||}{1-2||R^{-1}|| ||\delta R||} [||\delta R|| ||x|| + \epsilon'_1 ||b||]$$

provided

$$||R^{-1}|| ||\delta R|| < \frac{1}{2}$$

and $m = ||\delta R||.$

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

ORIGINATING ACTIVITY (Corporate author) Department of Computer Science University of Illinois at Urbana-Champaign Urbana, Illinois 61801	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP
---	---

REPORT TITLE

AN ERROR ANALYSIS OF SOLUTIONS TO SPARSE LINEAR PROGRAMMING PROBLEMS

DESCRIPTIVE NOTES (Type of report and inclusive dates)

Research Report

AUTHOR(S) (First name, middle initial, last name)

Raymond J. Lermitt

REPORT DATE September 17, 1969	7a. TOTAL NO. OF PAGES 69	7b. NO. OF REFS 9
6. CONTRACT OR GRANT NO. 46-26-15-305 b. PROJECT NO. US AF 30(602)4144 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) DCS Report No. 351 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	

0. DISTRIBUTION STATEMENT

Qualified requesters may obtain copies from DCS.

1. SUPPLEMENTARY NOTES NONE	12. SPONSORING MILITARY ACTIVITY Rome Air Development Center Griffiss Air Force Base Rome, New York 13440
--------------------------------	--

3. ABSTRACT

This thesis discusses the round-off errors incurred in solving Linear Programming problems by the Revised Simplex Method, using the Product Form of the Inverse - the most usual method of solution. The matrices involved are assumed to be sparse as is usually the case.

Also discussed are the reinversion of the basis matrix and iterative refinement of the solution obtained.

Security Classification

14

KEY WORDS

LINK A

L W K B

LINK C

ROLE

WT

ROLE

WT

ROLE

WY

Matrix inversion

Security Classification

UNIVERSITY OF ILLINOIS-URBANA

510 84 IL6R no. C002 no.349-354(1969
Report /



3 0112 088398737